UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/654,743	09/03/2003	Shmuel Hen	P16176	5235

46915 7590 11/01/2007
KONRAD RAYNES & VICTOR, LLP.
ATTN: INT77
315 SOUTH BEVERLY DRIVE, SUITE 210
BEVERLY HILLS, CA 90212

EXAMINER	
VERDI, KIMBLEANN C	
ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
11/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

Office Action Summary	Application No.	Applicant(s)	
	10/654,743	HEN ET AL.	
	Examiner	Art Unit	
	KimbleAnn Verdi	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 August 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) *1-9,11-22 and 24-34, 36-39* is/are rejected.

7)☒ Claim(s) *10,23 and 35* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This office action is in response to the Amendment filed on August 13, 2007. Claims 1-39 are pending in the current application. All previously outstanding objections and rejections to the Applicant's disclosure and claims not contained in this Action have been respectfully withdrawn by the Examiner hereto.

Response to Arguments

1. Applicant's arguments filed on August 13, 2007 have been fully considered but they are not persuasive. In response to the Non-Final Office Action dated April 11, 2007, applicant argues in regards to claims 1, 3, 5, 14-17, 19, 27, 29, 31, 39:

(1) **Nowhere do the cited cols. 9-10 (of Ogawa) disclose the requirement that a kernel module execute a kernel thread to call device driver function in a kernel context, where the device driver functions are those device driver functions capable of being called by system calls in the user context** (in regards to claims 1, 15, and 27, page 10, lines 26-28, and page 11 line 1).

In response to argument (1), examiner respectfully disagrees and notes that claims 1, 15, and 27 do not recite the feature of "a kernel module execute a kernel thread to call device driver function in a kernel context, where the device driver functions are those device driver functions capable of being called by system calls in the user context"; therefore applicant's argument is not persuasive because claims 1, 15, and 27 do not require the limitation of "a kernel module execute a kernel thread to call device driver function in a kernel context, where the device driver functions are those device driver functions capable of being called by system calls in the user context".

Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

(2) **Nowhere do the cited sections of Corbet teach or suggest that the kernel thread access device information from the device** (in regards to claims 3, 17, and 29, page 11, lines 28-29).

In response to argument (2), examiner respectfully disagrees and notes that Ogawa as modified by Corbet teaches accessing, with one kernel thread, device information from the device (kernel thread uses function aread, col. 11, lines 50-57 of Ogawa); and

buffering the accessed device information (stored in net_device structure as module in kernel, e.g. capabilities data in kernel, Fig. 2-1, linking a module to the kernel, Chapter 2, pages 1 and 2, data in structured inserted by driver for new interface into global list of network devices, chapter 14, section 14.2.1., lines 7-8).

(3) **Although the cited Corbet discusses methods for transmitting packets, nowhere is there any teaching or suggestion that a kernel thread access buffered device information periodically and independently of kernel module requests for the device information** (in regards to claims 5, 19, and 31, page 12, lines 12-15).

In response to argument (3), examiner respectfully disagrees and notes that Ogawa as modified by Corbet teaches wherein the kernel thread accesses buffered device information and independently of kernel module requests for the device

information periodically (kernel thread uses function aread, col. 11, lines 50-57 of Ogawa) (hard_start_xmit method initiates transmission of a packet utilizing the net_device structure (e.g. capabilities data in kernel, Fig. 2-1, hard_start_xmit method accesses capabilities data in kernel independently of kernel module request for device information to transmit packet) chapter 14, section 14.3.2.2, page 1, lines 22-26 of Corbet).

(4) **This cited pg. 21 does not teach the claim requirement of disabling any higher priority contexts capable of accessing the device information** (in regards to claims 14, 26, and 39, page 13, lines 11-13).

In response to argument (4), examiner respectfully disagrees and notes that claims 14, 26, and 39 do not recite the feature of "disabling any higher priority contexts capable of accessing the device information"; therefore applicant's argument is not persuasive because claims 14, 26, and 39 do not require the limitation of "disabling any higher priority contexts capable of accessing the device information". Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Specification

2. The abstract of the disclosure is objected to because the period is missing at the end of the last sentence. Correction is required. See MPEP § 608.01(b).

Allowable Subject Matter

3. Claim 10 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claims 1-39 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Examiner notes that that the corresponding support on pages 3-4 of the specification paragraph [0009] is inconsistent with the claim language in claims 1, 15, and 27. For purposes of examination the limitation of "wherein the device driver functions are capable of being invoked system calls from applications operating in a user context" is interpreted by the examiner as "wherein the device driver functions are capable of being invoked by system calls from applications operating in a user context", as in accordance with the applicant's specification.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1-2, 15-16, and 27- 28 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 6,754,736 B1 to Ogawa et al. (hereinafter Ogawa).

8. As to claim 1, Ogawa teaches a method for communicating with a device, comprising:

executing a kernel module in a memory (driver 20, Fig. 1 operating in Kernel 50, Fig. 2, Kernel resident in memory);

executing at least one kernel thread (kernel thread T0, Fig. 9) in the memory (Kernel 50, Fig. 2, Kernel resident in memory) to handle calls to device driver functions for the kernel module (calls input/output function for the kernel device driver, col. 9, lines 16 and 66), wherein the device driver functions are capable of being invoked by system calls from applications operating in a user context (col. 11, lines 50-58); and

executing, with the at least one kernel thread (kernel thread T0, Fig. 9), calls to device driver functions (T0 calls input/output function to activate asynchronous thread A0, col. 9, lines 66-67) for the kernel module (driver 20, Fig. 1, device driver, col. 9, line 16) running in a kernel context (T0 and A0 runs in kernel space, Fig. 9).

9. As to claim 2, Ogawa teaches the method of claim 1, wherein the kernel module spawns at least one kernel thread (kernel thread T0 activates asynchronous thread A0 (other kernel threads, col. 9, lines 22-24)) to execute the calls to the device driver functions for the kernel module (T0 calls input/output function to activate asynchronous thread A0, col. 9, lines 66-67, asynchronous threads (An), execute input/out request, 64, Fig. 10).

10. As to claim 15, Ogawa teaches a system, comprising:

a network device (network interface controller 22, Fig. 1, corresponds to one of the network adaptors A0-An, Fig. 12, col. 4, lines 46-51, each network adaptor represent one network device in a system, col. 1, lines 59-63);

a memory (main storage device consisting of rom and ram, 82, Fig. 20);

a processor executing code to perform (cpu, 81, Fig.20 executes program and data of Fig. 21):

(i) execute a network device driver (Communication Management Unit 17, Fig. 1) in memory to control the network device (directly controls a network interface controller 22, Fig. 1, col. 20, lines 4-11);

(ii) execute a kernel module in the memory (driver 20, Fig. 1 operating in Kernel 50, Fig. 2, Kernel resident in memory);

(iii) execute at least one kernel thread (kernel thread T0, Fig. 9) in the memory (Kernel 50, Fig. 2, Kernel resident in memory) to handle calls to device driver functions for the kernel module (calls input/output function for the kernel device driver, col. 9, lines 16 and 66); and

(iv) execute, with the at least one kernel thread (kernel thread T0, Fig. 9), calls to device driver functions (T0 calls input/output function to activate asynchronous thread A0, col. 9, lines 66-67) for the kernel module (driver 20, Fig. 1, device driver, col. 9, line 16) running in a kernel context (T0 and A0 runs in kernel space, Fig. 9).

11. As to claim 16, this claim is rejected for the same reasons as claim 2 since claim 19 recites the same or equivalent invention, see the rejection to claim 2 above.

12. As to claims 27-28, these claims are rejected for the same reasons as claims 1-2 respectively, since claims 27-28 recite the same or equivalent invention, see the rejections to claims 1-2 above.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. Claims 3-10, 13, 17-23, 25, 29-35, and 38 rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,754,736 B1 to Ogawa et al. (hereinafter Ogawa) in view of "Linux Device Drivers, 2nd Edition" by J. Corbet and A. Rubini (hereinafter Corbet).

15. As to claim 3, Ogawa teaches the method of claim 1 further comprising:

accessing, with one kernel thread, device information from the device (kernel thread uses function aread, col. 11, lines 50-57).

Ogawa does not explicitly disclose buffering the accessed device information.

However Corbet teaches buffering the accessed device information (stored in net_device structure as module in kernel, e.g. capabilities data in kernel, Fig. 2-1, linking a module to the kernel, Chapter 2, pages 1 and 2, data in structured inserted by driver for new interface into global list of network devices, chapter 14, section 14.2.1., lines 7-8).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the kernel driver of Ogawa with the teachings of a net_device structure from Corbet because this feature would have a mechanism for supporting a number of administrative tasks, such as setting addresses, modifying transmission parameters, and maintaining traffic and error statistics (Chapter 14, page 1, lines 28-29 of Corbet).

16. As to claim 4, Ogawa as modified by Corbet teaches the method of claim 3, wherein a kernel module function requests device information (get stats function part of the device methods of kernel- driver interface, chapter 14, section 14.3.2, lines 4-5, chapter 14, section 14.3.2.2, page 2, lines 21-25 of Corbet), further comprising: in response to the request for the device information, accessing the buffered device information (get stats function accesses the net_device structure (e.g. capabilities data in kernel, Fig. 2-1) to return the net_device_stats, chapter 14, section 14.3.2.2., page 2, lines 21-25 of Corbet).

17. As to claim 5, Ogawa as modified by Corbet teaches the method of claim 1, wherein the kernel thread accesses buffered device information and independently of kernel module requests for the device information periodically (kernel thread uses

function aread, col. 11, lines 50-57 of Ogawa) (hard_start_xmit method initiates transmission of a packet utilizing the net_device structure (e.g. capabilities data in kernel, Fig. 2-1, hard_start_xmit method accesses capabilities data in kernel independently of kernel module request for device information to transmit packet) chapter 14, section 14.3.2.2, page 1, lines 22-26 of Corbet).

18. As to claims 6, Ogawa as modified by Corbet teaches the method of claim 1, further comprising:

buffering a parameter list (e.g. multicast list, stored in net_device structure as module in kernel, e.g. capabilities data in kernel, Fig. 2-1, linking a module to the kernel, Chapter 2, pages 1 and 2 of Corbet); and

setting device parameters in the buffered parameter list to values provided by kernel module functions (e.g. multicast list, kernel uses the method dev->set_multicast_list to notify driver whenever the list of valid multicast addresses is changed, dev->set_multicast_list, chapter 14, section 14.13, page 2, lines 1-3, and section 14.13.1, lines 5-11 of Corbet).

19. As to claim 7, Ogawa as modified by Corbet teaches the method of claim 6, further comprising: setting a flag indicating that the kernel thread needs to set parameters at the device to device parameter values set in the parameter list (whenever dev->flags is modified (e.g. IFF_PROMISC) the function kernel uses the method dev->set_multicast_list is invoked to reprogram the hardware filter chapter 14, section 14.13.1, lines 8-9 of Corbet).

20. As to claim 8, Ogawa as modified by Corbet teaches the method of claim 6, further comprising:

spawning (e.g. fork) a kernel thread (kmod.thread is called by kernel to access modules, chapter 11, section 11.1, lines 11-14 of Corbet) to set device parameters to parameter values buffered in the parameter list (kmod thread would use kernel- driver interface to invoke method dev->set_multicast_list to notify driver whenever the list of valid multicast addresses is changed, dev->set_multicast_list, and reprogram hardware filter, chapter 14, section 14.13, page 2, lines 1-3, section 14.13.1, lines 5-11, section 14.3.2, lines 4-5, and section 14.3.2.2, page 2, lines 21-25 of Corbet).

21. As to claim 9, Ogawa as modified by Corbet teaches the method of claim 7, wherein the kernel thread spawned (kmod.thread is called by kernel to access modules, chapter 11, section 11.1, lines 11-14 of Corbet) to set device parameter values (kmod thread would use kernel- driver interface to invoke method dev->set_multicast_list to notify driver whenever the list of valid multicast addresses is changed, dev->set_multicast_list, and reprogram hardware filter, chapter 14, section 14.13, page 2, lines 1-3, section 14.13.1, lines 5-11, section 14.3.2, lines 4-5, and section 14.3.2.2, page 2, lines 21-25 of Corbet) processes the parameter list to locate buffered parameter values (e.g. dev->flag modified) and set the device parameters to the buffered parameter values (e.g. kernel- driver interface to invokes method dev>set_multicast_list chapter 14, section 14.13, page 2, lines 1-3, section 14.13.1, lines 5-11, section 14.3.2, lines 4-5, and section 14.3.2.2, page 2, lines 21-25 of Corbet).

22. As to claims 10, Ogawa as modified by Corbet teaches the method of claim 7, wherein the kernel thread processes the parameter list by further performing:

applying a lock on information in the parameter list including the located buffered parameter values (kmod thread would use kernel- driver interface to set spinlock_t xmit_lock on the sk_buff of the net_device structure, chapter 14, section 14.3.2.3., page 2, lines 12-18, chapter 14.5, packet transmission, lines 29 of Corbet);

after applying the lock, copying the parameter values from the parameter list to a temporary buffer (hard_start_transmit method called by kernel to put data packet (sk_buff) on out going queue (buffer), chapter 14, section 14.5, packet transmission, lines5-7 of Corbet), wherein the device parameters are set to the parameter values from the parameter list in the temporary buffer (copy of sk_buff is passed as parameter to net_device in snull_tx method call, chapter 14, section 14.5, packet transmission, line 29 of Corbet) ; and

releasing the lock after copying the parameter values from the parameter list to the temporary buffer (hard_start_xmit function releases spinlock (xmit_lock) on function return, chapter 14, section 14.5.1, lines 2-3 of Corbet).

23. As to claim 13, Ogawa as modified by Corbet teaches the method of claim 10, further comprising:

after releasing the lock, executing device driver functions (hard_start_xmit function releases spinlock (xmit_lock) on function return, and function can be called again, chapter 14, section 14.5.1, lines 2-3 of Corbet) to configure the device with the parameter values in the temporary buffer (copy of sk_buff is passed as parameter to

net_device in snull_tx method call, chapter 14, section 14.5, packet transmission, line 29 of Corbet).

24. As to claims 17-21, these claims are rejected for the same reasons as claims 3-7 respectively, since claims 17-21 recite the same or equivalent invention, see the rejections to claims 3-7 above.

25. As to claims 22-23, these claims are rejected for the same reasons as claims 9-10 respectively, since claims 22-23 recite the same or equivalent invention, see the rejections to claims 9-10 above.

26. As to claim 25, this claim is rejected for the same reasons as claim 13 since claim 25 recites the same or equivalent invention, see the rejection to claim 13 above.

27. As to claims 29-33, these claims are rejected for the same reasons as claims 3-7 respectively, since claims 29-33 recite the same or equivalent invention, see the rejections to claims 3-7 above.

28. As to claims 34-35, these claims are rejected for the same reasons as claims 9-10 respectively, since claims 34-35 recite the same or equivalent invention, see the rejections to claims 9-10 above.

29. As to claim 38, this claim is rejected for the same reasons as claim 13 since claim 38 recites the same or equivalent invention, see the rejection to claim 13 above.

30. Claims 11, 12, 14, 24-26, 36, 37, and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,754,736 B1 to Ogawa et al. (hereinafter Ogawa) in view of "Linux Device Drivers, 2nd Edition" by J. Corbet and A. Rubini (hereinafter

Corbet) as applied to claims 10, 23, and 35 above, and further in view of "Synchronization in Portable Device Drivers" by Stein J. Ryan (hereinafter Ryan).

31. As to claim 11, Ogawa as modified by Corbet does not explicitly teach disabling higher priority contexts before locking the parameter list; and

enabling the higher priority contexts after releasing the lock on the parameter list.

However Ryan teaches disabling higher priority contexts before locking the parameter list (device driver disables first and second stage interrupt processing when using spinlock, page 21, right column, lines 10-11); and

enabling the higher priority contexts after releasing the lock on the parameter list, (releasing spinlock enables interrupt processing, and potentially executes a queued second stage handler, page 21, left column, section 4.2, lines 2-3, and right column lines 25-26).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the spinlock of Ogawa as modified by Corbet with the teachings of a spinlock from Ryan because this feature would have further provided a mechanism for enabling and disabling interrupt processing to prevent deadlocks (page 21, section 4.2, lines 2-3 of Ryan).

32. As to claim 12, Ogawa as further modified teaches the method of claim 11, wherein the higher priority context comprises a bottom half or Interrupt Request (IRQ) context (e.g. hardware context) (first stage interrupt processing is done by manipulating the hardware interrupt mask of the interrupt controller, page 21, right column lines 1 and 29-30 of Ryan).

33. As to claim 14, Ogawa as further modified teaches the method of claim 1, further comprising:

initiating, with the kernel module, an access request with respect to device information (get stats function part of the device methods of kernel- driver interface, chapter 14, section 14.3.2, lines 4-5, chapter 14, section 14.3.2.2, page 2, lines 21-25 of Corbet);

disabling any higher priority contexts capable of accessing the device information (device driver disables first and second stage interrupt processing when using spinlock, page 21, right column, lines 10-11 of Ryan);

obtaining a lock for the device information subject to the access request values (kmod thread would use kernel- driver interface to set spinlock_t xmit_lock on the sk_buff of the net_device structure, chapter 14, section 14.3.2.3., page 2, lines 12-18, chapter 14.5, packet transmission, lines 29 of Corbet);

providing the kernel module access to the device information (device methods for network interface (adapter) used to obtain device information to the kernel, chapter 14, section 14.3.2.2., line 1 of Corbet);

releasing the lock (releasing spinlock enables interrupt processing, page 21, left column, section 4.2, lines 2-3 of Ryan); and

enabling all higher priority contexts that were disabled (releasing spinlock enables interrupt processing, and potentially executes a queued second stage handler, page 21, left column, section 4.2, lines 2-3, and right column lines 25-26 of Ryan).

34. As to claim 24, this claim is rejected for the same reasons as claim 11 since claim 24 recites the same or equivalent invention, see the rejection to claim 11 above.

35. As to claim 25, this claim is rejected for the same reasons as claim 13 since claim 25 recites the same or equivalent invention, see the rejection to claim 13 above.

36. As to claim 26, this claim is rejected for the same reasons as claim 14 since claim 26 recites the same or equivalent invention, see the rejection to claim 14 above.

37. As to claims 36-37, these claims are rejected for the same reasons as claims 11-12 respectively, since claims 36-37 recite the same or equivalent invention, see the rejections to claims 1-12 above.

38. As to claim 38, this claim is rejected for the same reasons as claim 13 since claim 38 recites the same or equivalent invention, see the rejection to claim 13 above.

39. As to claim 39, this claim is rejected for the same reasons as claim 14 since claim 39 recites the same or equivalent invention, see the rejection to claim 14 above.

Conclusion

40. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KimbleAnn Verdi whose telephone number is (571) 270-1654. The examiner can normally be reached on Monday-Friday 7:30am-5:00pm EST..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571) 272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

KV
October 19, 2007

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER